Basic ATM Controller

Purpose:

- This module covers the basic operation of the FCC in ATM mode dealing with how the cells are received and transmitted. It describes address mapping, pace control, channel data handling, buffer pools, and interrupts.

Objectives:

- After completing this section you will have a basic understanding of the functionality of the ATM Controller and a description of the activity within the controller as packets are processed.

Contents:

- This section contains a basic overview of the ATM Controller including the memory structures and parameters with examples of how the FCC transmits ATM cells. There is also a detailed description of the various registers towards the end of the presentation.

Learning Time:

- There are 38 pages and 14 reference pages which will take approximately 65 minutes.

This module covers the basic operation of the FCC in ATM mode dealing with how the cells are received and transmitted. It describes address mapping, pace control, channel data handling, buffer pools, and interrupts.

After completing this section you will have a basic understanding of the functionality of the ATM Controller and a description of the activity within the controller as packets are processed.

This section contains a basic overview of the ATM Controller including the memory structures and parameters with examples of how the FCC transmits ATM cells. There is also a detailed description of the various registers towards the end of the presentation.

There are 38 pages and 14 reference pages which will take approximately 65 minutes.

Connection Options

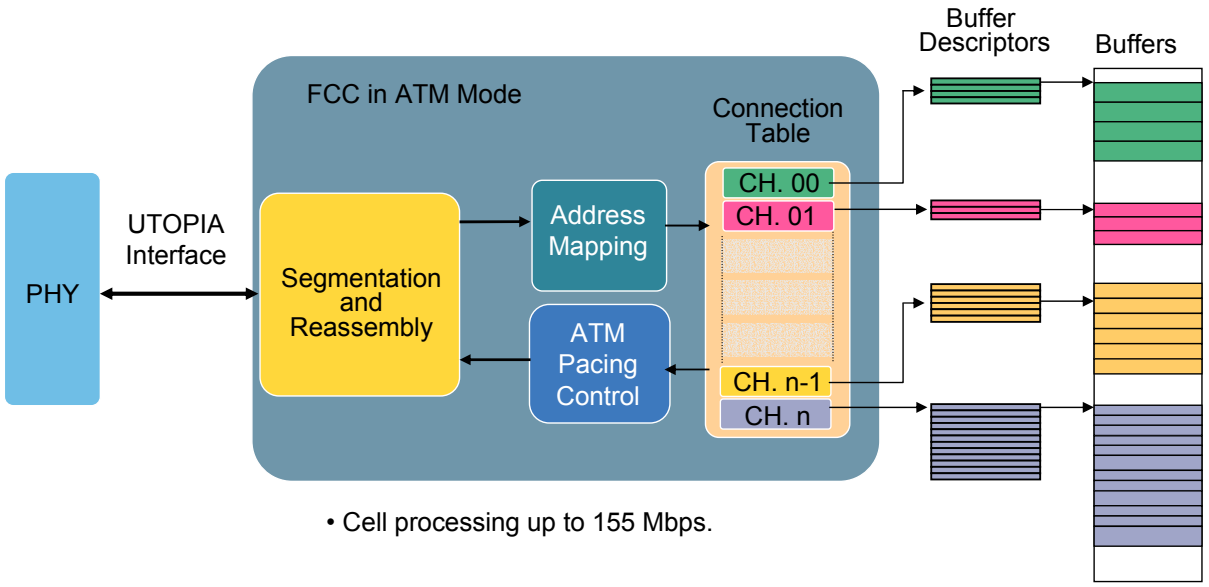
- The FCCs can only be connected for ATM using the UTOPIA interface

FCC 1	16-bit master or slave or 8-bit master or slave
FCC2	8-bit master or slave
FCC3	has no ATM capability

The ATM protocol can only be connected via a UTOPIA interface which is available on FCC1 and FCC2. FCC1 allows 16-bit data connections and both FCC1 and 2 allow 8-bit connections. FCC3 cannot handle ATM at all.

I Controller

- The FCC controller can transmit and receive data using ATM to a physical device with a UTOPIA interface.



- ATM data can also be transferred synchronously using the ATM-to-TDM bridging mode.

The basic functionality of the controller is to receive ATM cells from the PHY via the utopia interface, utilize the cell header to determine what channel related to this FCC the cell belongs to, and transfer the data from the cell to the next available location in the appropriate buffer. The reverse is also true. When the PHY indicates that it has room for a cell to transmit the controller must decide which is the next channel to be transmitted, locate the buffer containing the required cell, add the appropriate header to the cell, and transfer the cell to the PHY.

This is a very simple description of the activity and as you progress through the course the process will be described in more detail. The process of segmentation and re-assembly is simply the transfer of the required data between the cells and buffers, where the cell contains the fixed block of data between 46 and 48 bytes depending on the adaptation layer used, and the buffers contain the user data of whatever size it is.

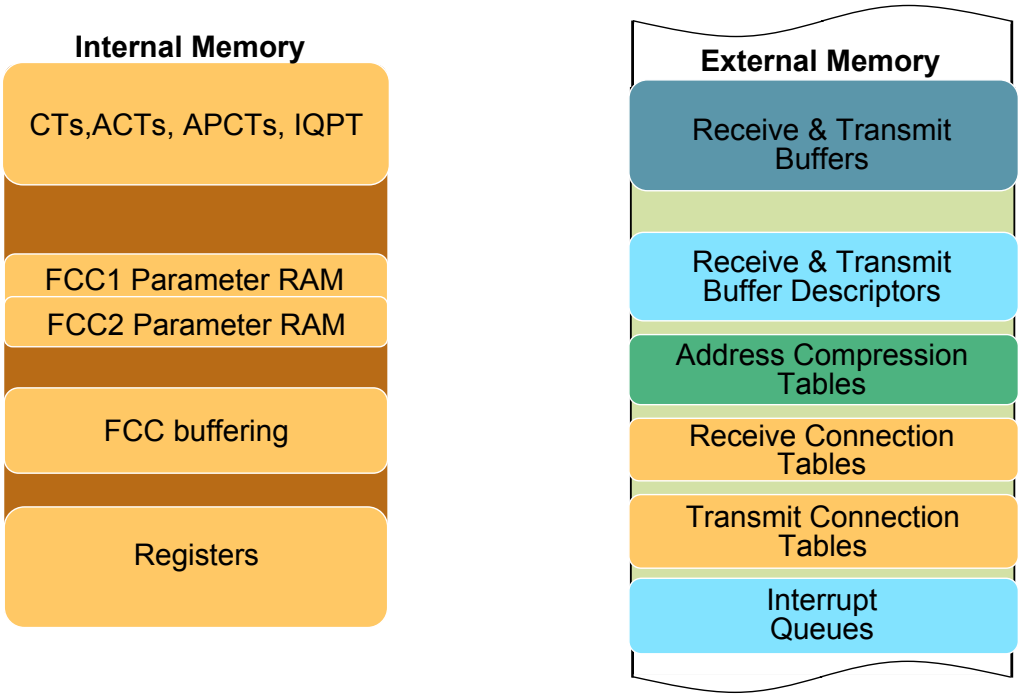
Address mapping is the process of relating the address indicators in the cell header to the appropriate channel, and from there selecting the correct buffer. Buffers are handled in the same way as for other protocols, where the buffer control is defined in a buffer descriptor. In this case each channel has it's own buffer descriptors which are accessed via the channel connection table entry. The channel connection tables provide the similar types of information that for the simpler protocols is defined in a simple set of parameters. Because with ATM, there can be very many channels, this information is handled by the Connection table.

The transmitter is complicated by the scheduling of channels for transfer. Because ATM has the ability to mix data of varying priority and quality of service it is necessary for the pace control function to determine which channel cell must be scheduled to be sent to the transmitter at any given time. To allow the high data rates expected for the service this scheduling is done in advance to have the cells located in the FIFO, ready to transfer to the PHY when requested. To relate this to the general aspect of ATM, one channel could be for video data, possibly a streaming movie, one could be a telephone service, another an internet connection, while one could be a computer file transfer. All could be utilizing very different service schemes from low delay, constant rate, to bursts of data with no timing constraints.

The FCC can handle ATM data up to 155 megabits per second, although other functions handled by the PowerQUICC II must be taken into consideration as well as the adaptation layer used. ATM to TDM direct connections are possible using a special bridging mechanism.

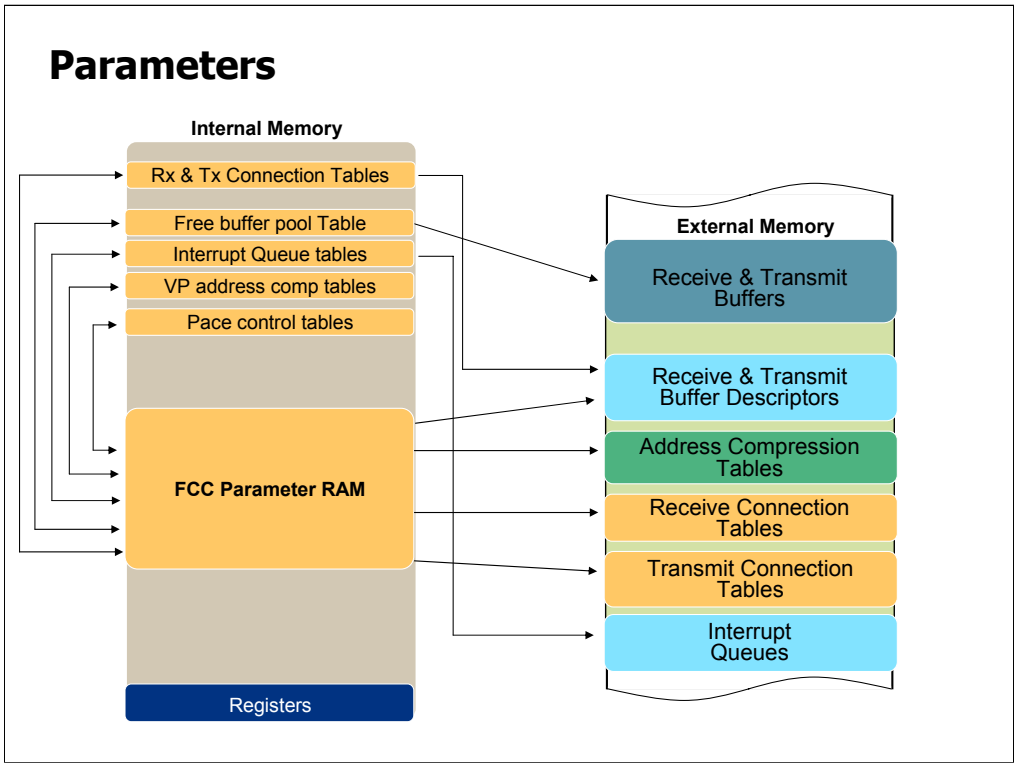
ATM memory structures

- As for all CPM functionality, the RISC processor requires information in DPR to enable it to control the protocol and access memory. In the case of ATM there could be too much information to be totally contained in DPR and so further areas in external memory are used.



The risk processor of the CPM is controlling the FCC for this function and so requires all of the same type of information as for other protocols. However, in ATM mode there are many more parameters needed for handling all the different functions necessary. On the left is the internal dual-port-ram map showing the four basic areas. The FCC parameters which hold the base pointers to all the other parameters required have pre-defined locations, which is necessary so that the communications processor knows where to look.

For the ATM function to operate parameters are required for each channel. There are address compression tables for the receiver to decide which channel any given cell relates to, connection tables for both receiver and transmitter for defining all the protocol and data routing information for each channel, pace control tables for directing the scheduling of cell to transmit, Interrupt queues for both receive and transmit, and buffer descriptors and buffers for the data. Some of these parameters are in DP RAM, some in external memory. FCC buffering is an intermediate buffer to improve data throughput and where some processing is handled by the CP and the register area is where the basic functionality of the FCC is defined.



Here are the basic parameters used. In FCC, parameter RAM are pointers to receiver and transmitter connection tables in DP RAM, which are used for the lower numbered channels zero through 255. The pointers to this space are simply 16-bit offsets from the top. Where possible, high priority, high frequency channels should be allocated low numbers so that they will be controlled from this space where there is less delay in handling them. The connection tables provide all the channel control information and buffer descriptor pointers.

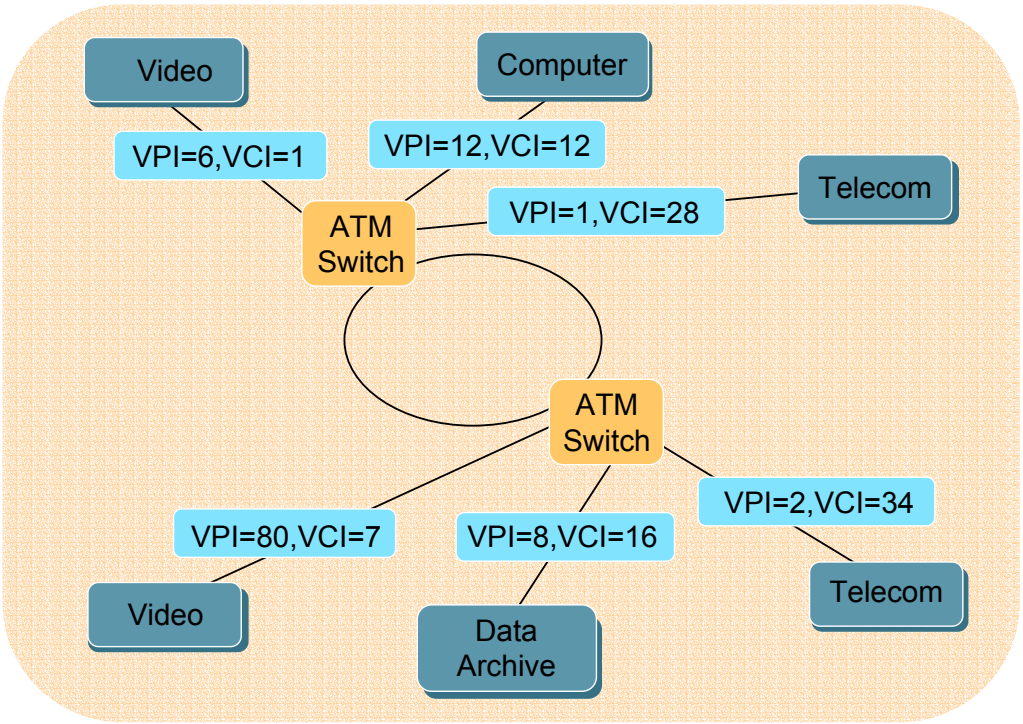
The connection tables for the higher numbered channels must be in external memory and so there are FCC parameters pointing to this space. The actual table entry is calculated through the channel number. There is an FCC parameter pointing to the VP address compression table in DP RAM, used in the recognition of which channel the incoming cell relates to and another to the VC address compression table in external memory which is also used in the process, as it is a two part function. These are only required if CAM addressing is not used. Pace control tables are all in DP RAM because they are required to be very quick access to support quality of service, and are used to schedule cell delivery to the FIFO.

Events occur related to channels and so it's not possible to have a simple event register to handle them. In this case the event register is used to indicate that one of a few types of event has occurred and the specific event information is loaded into an interrupt queue. The queue is located in external memory, and an interrupt queue parameter table is in DP RAM, to identify the base address of the queue and entry tracking information.

To improve the efficiency of memory allocation global buffer allocation can be utilized. This is a method of pre-defining a number of buffers in memory without allocating them to specific channels. As any channel requires a new buffer it is obtained from the free buffer pool. In this way a large memory allocation for buffers may not be required for channels which may not need it. An FCC parameter points to the free buffer pool parameter table in DP RAM. In all cases the FCC parameters are 16-bit offsets to locations in DP RAM one, and 32-bit absolute addresses to external memory.

VPI and VCI

- VPI, virtual path identifier, and VCI, virtual circuit identifier, provide the addressing on an ATM network.

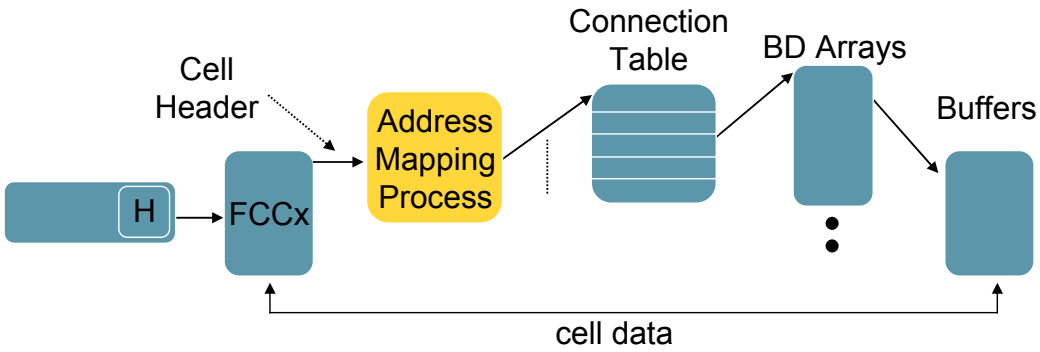


VPI is the virtual path identifier and can be considered like the outer case of a multi-core cable. It identifies the major route of the data. VCI is the virtual circuit identifier and can be considered like an inner core of the cable. These values are used to route the cell data to the required destination. However, the difficult concept here is that when the cell is transmitted, these values are not defining the final destination but simply the first stage routing. A very brief explanation of what happens is that before any data is transmitted the system has to go through a form of negotiation to find the best routing for the cell and in the process each node or ATM switch must set up a table for the VPI and VCI values, and priority and quality of service for the transfer. This is all handled by higher level applications software outside the FCC operation and so is not a function of this training. The VPI and VCI is added into the header of the cell when transmitted, and the receiver examines these values to determine which channel the cell belongs to.

In this very simple example each different data stream is allocated a value which will enable the ATM switch to determine the channel to send the cell to. The channel is a simple method of defining how to reassemble related cells into predefined buffers. This reassembled data can then be retransmitted onto the next stage with another set of identifiers that will get it to the appropriate channel at the next stage. Eventually the last stage will be the location determined by the pre-transmission negotiation phase. At each stage the VPI and VCI values are unrelated to the previous ones, and are subject to the actual routing allocation from the switch routing tables. As far as the FCC is concerned these values are given, supplied from parameters initialized by the user.

Address Mapping

- Address mapping is the operation that occurs to determine the connection through which the data from an incoming frame should be moved.



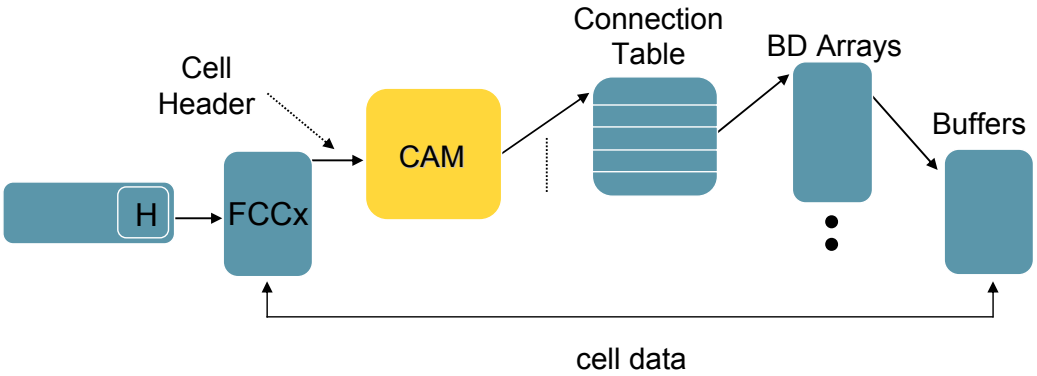
Type	GMODE[ALM]
External CAM	0
Address Compression	1

- GMODE is a parameter in ATM parameter RAM (see page 29-39 in user's manual).

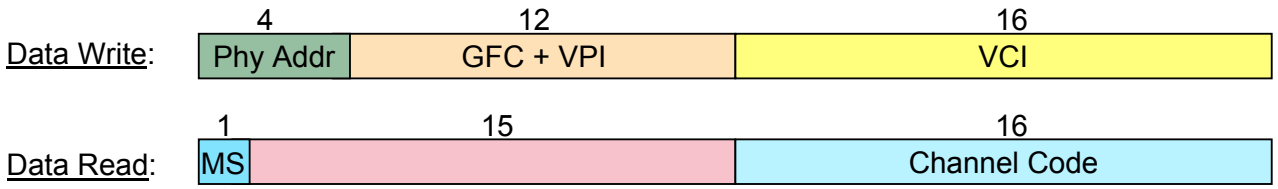
The method of processing received cells is VPI and VCI are used from the cell header to perform address mapping. Two options are available which are selected in an FCC parameter called Gmode. The ALM bit determines whether address mapping utilizes an external CAM for hardware recognition of channels, or address compression which is a software method of converting the identifiers to a channel number. Once the channel has been determined the channel's connection table entry is used to provide all the necessary channel data handling information, access the current or next buffer descriptor to transfer the cell data to the buffer.

CAM Mapping

When a CAM is used for address mapping, the cell header is written to the CAM and a subsequent read obtains the channel code.



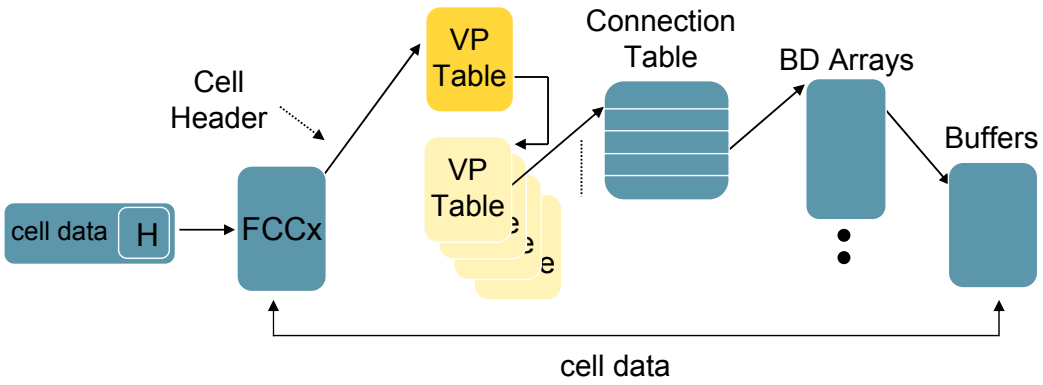
Data Formats



When a wide range of addresses are to be used a CAM is the most efficient method of determining the channel the cell belongs to. In this case the CP automatically writes the first 32-bits of the header, which includes the VPI, VCI and other header information which could be utilized for addressing, to the CAM and then reads the response. The response is the channel number and the MS bit which indicates if the address is valid. If the MS bit is clear, then the channel number is valid and the cell is accepted. If it is set, then this is considered a mis-inserted cell and is discarded. To see an example of a CAM connection, click on the CAM in the diagram.

Address Compression Mapping

- Address compression mapping is a dual-level, address lookup method. It covers the available address range while using minimum memory space.



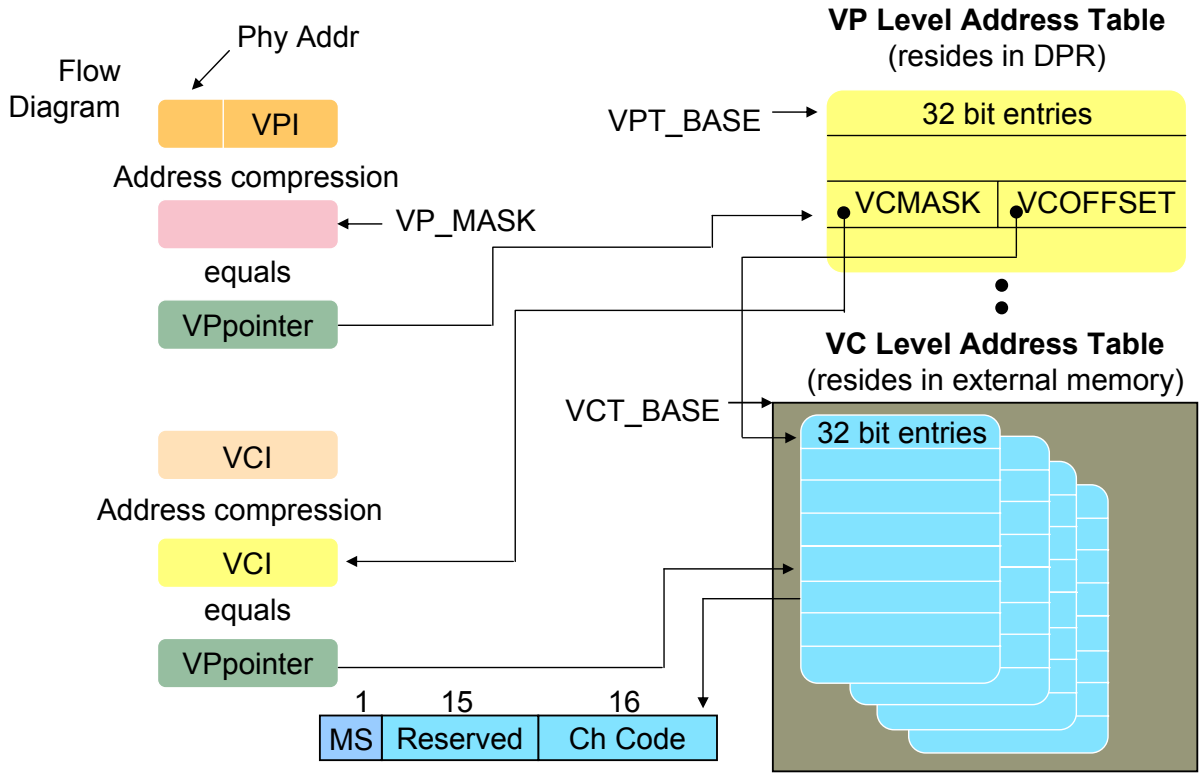
- VP (first level) is the Virtual Path Table
- VC (second level) is the Virtual Connection Table

- The address compression mapping method is most valuable when you are expecting only some values for VPI/VCI. For handling all possible combinations, the CAM addressing method is better.

Address compression mapping is only useful when a relatively small number of channels are to be used. It uses a two level look-up table mechanism where the VPI provides an offset into the first level VP table, which is located in DP RAM and provides a pointer to the appropriate VC, table in external memory. The second level provides the channel number.

A specific consideration for this is how to keep the table allocation a reasonable size. At the very minimum, if only the VPI value is used, which is an 8 bit value, then the first level table must be 256 entries long. Each of those entries points to a second level table where the VCI provides an offset into that table which is 16-bits, so there would be a requirement for 256 tables each 65,536 entries long. Each entry consists of four bytes, so to set this up would require an enormous quantity of memory. Clearly this is not a practical solution. What is needed is a method of reducing the size of the tables needed for a particular application.

How the Address Compression Mechanism Operates



What must be considered is how many addresses are expected to be handled by this application. Often there will be a predefined range of VPI and VCI values used and so the range of table required to deal with them can be reduced by only monitoring those values expected. For this, a mask is used which is combined with the value received to determine which bits in the value are recognized and the rest are ignored.

FCC parameters provide the VP mask and the pointer to the VP table in DP RAM. It also provides a pointer to the base of the VC tables in external memory. A combination of the VPI value and mask provides an offset into the first level table. The VP table entry provides the VC mask and an offset to the appropriate VC table, and the combination of the VCI value and mask provides the offset into the VC table. That entry provides a value the same as that provided by a CAM, which indicates either a valid channel number or a mis-inserted cell.

The Mask

Simple example: Assuming that only the following values are expected to be received for an 8-bit address: 2, 3, 4, 65, 76, 138, 139

Address	Binary
2	0000 0010
3	0000 0011
4	0000 0100
65	0100 0001
76	0100 1100
138	1000 1010
139	1000 1011
OR'd	1100 1111

By reducing the number of bits to recognize and ignoring those not needed, the process can be simplified.

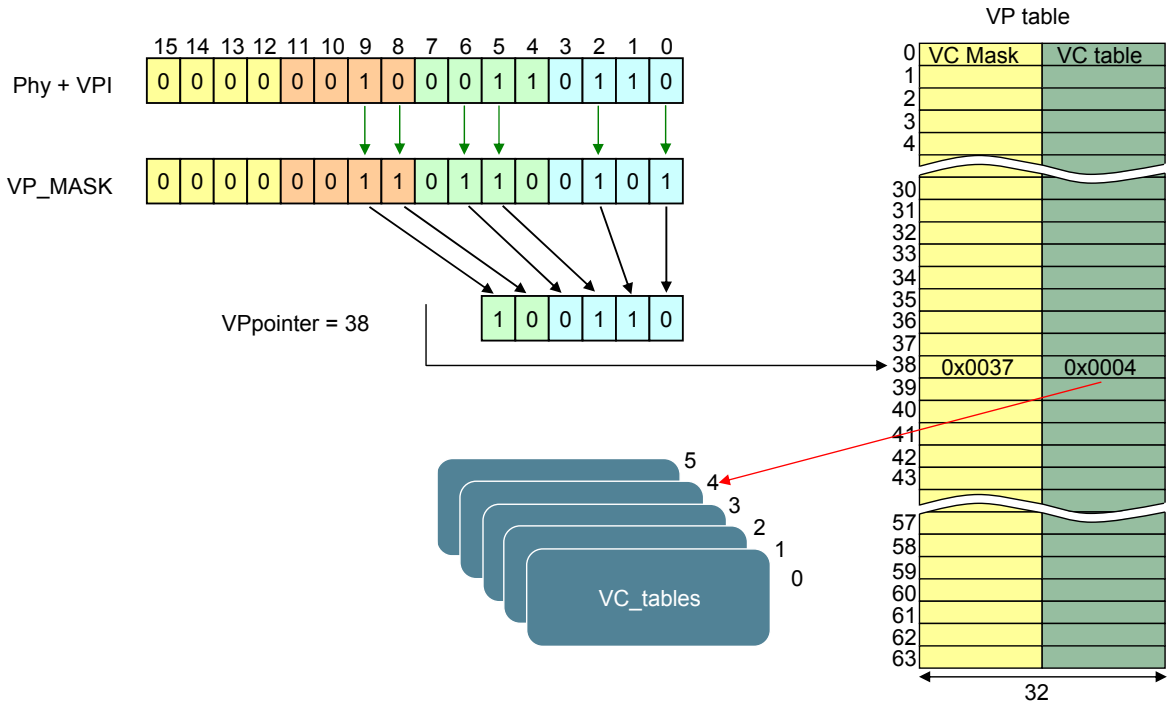
- Since only the highlighted bits are required to recognize these addresses a mask which enables us to ignore the other bits can significantly reduce the size of the table needed for address recognition.
- If all 8 bits were needed then a table of 256 entries is required, if only the bits indicated are used then 6 bits are used and the table is reduced to 64 entries.

Let's assume that this FCC only expects to receive addresses 2, 3, 4, 65, 76, 138, and 139. If the addresses are 8-bit then a table of 256 entries would be required to map all of the possibilities. Examining the bit patterns we see that we never receive a bit set in bits 4 and 5, and so these bits could be totally ignored by the recognition mechanism. That results in reducing the number of bits to compare to 6 which requires a table of 64 entries. There is obviously still a larger overhead than necessary but this is a significant improvement, reducing the amount of memory required for the table by 75 percent. Extending this amount of saving over two level tables is considerable.

Addresses Compression (VP Level Table)

Example Given the following parameters:

Phy + VPI = 0x236
VP_MASK = 0x365



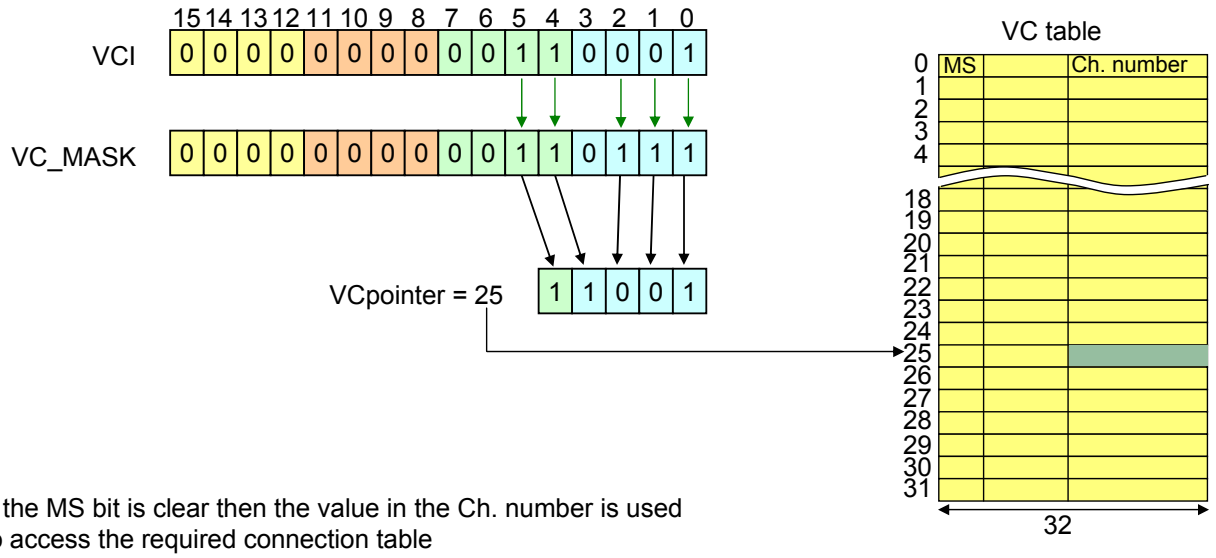
Only bits of the VPI, which align with bits set in the mask, are used to define an offset into the VP table. In this example, only bits 9, 8, 6, 5, 2, and 0 of the VPI are used to generate the offset into the VP table, generating a 6-bit value requiring a 64 entry table. When those bits are utilized, they generate the value decimal 38, which is the entry number into the VP table. That entry contains the mask to be used with the VCI and the offset to the required VC table. The process then continues to the VCI compression. Notice that there is the possibility of a different mask value for every VCI table so each table could be a different length.

Before moving on there is one further consideration here. Notice that in this example there is a bit set in the VPI, which is not matched by the mask. That represents a potential problem, since this will result in a table entry being used for a VPI that is not being compared and could result in an error. However, there is a mechanism which identifies this possibility and the user has the opportunity to discard this mis-inserted cell. This is the check unallocated bits function defined in the global mode entry of the FCC parameters.

Addresses Compression (VC Level Table)

Example Given the following parameters:

VCI = 0x31
VC_MASK = 0x37



If the MS bit is clear then the value in the Ch. number is used to access the required connection table

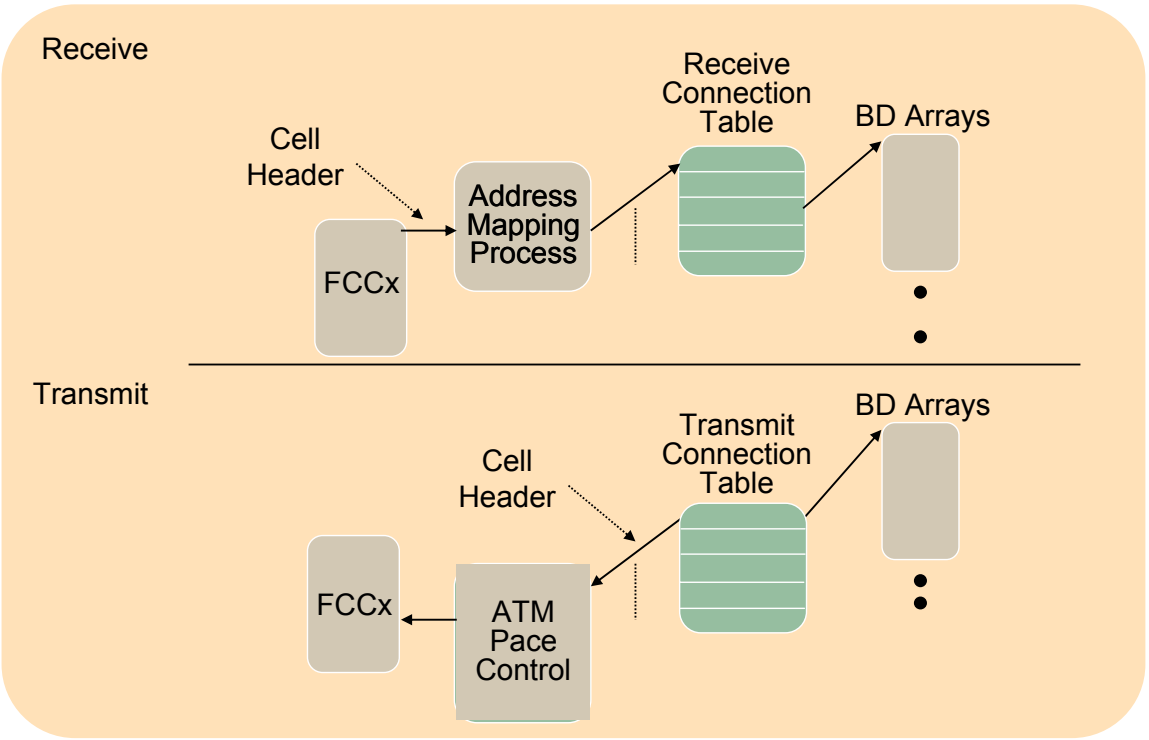
Exercise

Exactly the same mechanism is used for the VC level table. In this example only 5 bits of the VCI are used to access this table, requiring a table of 32 entries. In this case, only bits 5, 4, 2, 1, 0, are used from the VCI, which provides an offset into this table of 25. That entry provides the channel number that this cell relates to, and will direct the CP to that connection table for further information regarding the processing of this cell, provided that the MS bit indicates that it is valid.

The example on slide eleven defining the Mask indicates that only 7 values were expected, but a table of 64 entries is required. Clearly an indication of which entries are valid is required, which is the reason for the MS bit.

Connection Tables

- The connection tables hold channel configuration and temporary parameters for each channel, receive and transmit.

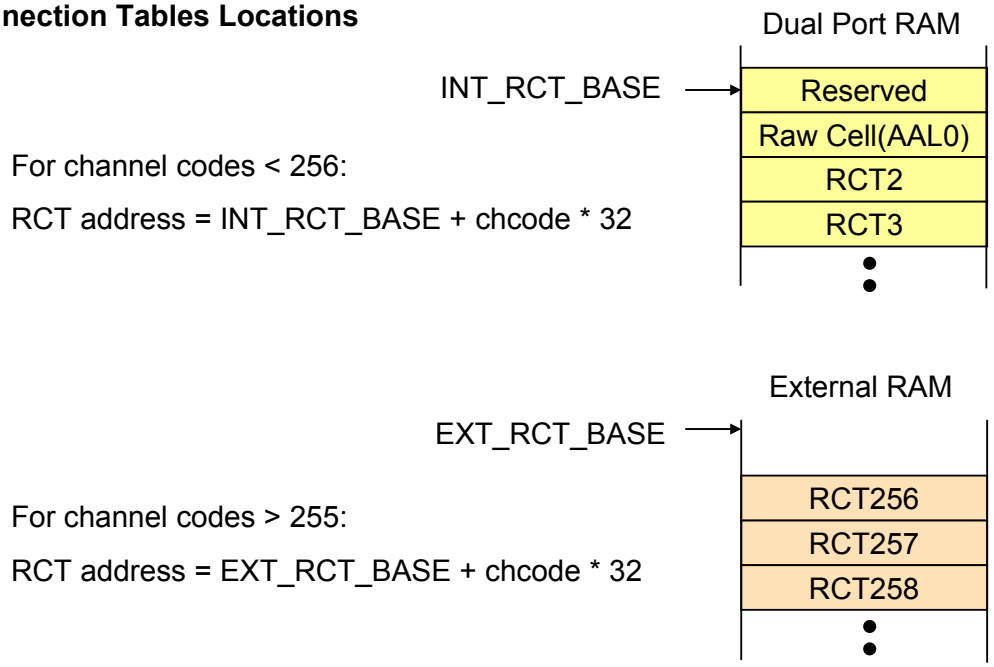


The connection tables contain configuration information for each channel, as well as temporary parameters for the CP to use for handling the channel. There are separate tables for both receive and transmit. The previous material indicates how the receiver determines the correct channel entry, and for the transmitter, the pace control mechanism defines which channel to use. That will be covered later. In all cases one of the parameters within each connection table entry is a pointer to the buffer descriptors to enable the transfer of data between FIFO and buffer.

Channel Code Pointer to the RCT

- The channel code resulting from the address lookup is a 16-bit value which determines the Receive Connection Table entry to be used.

Connection Tables Locations



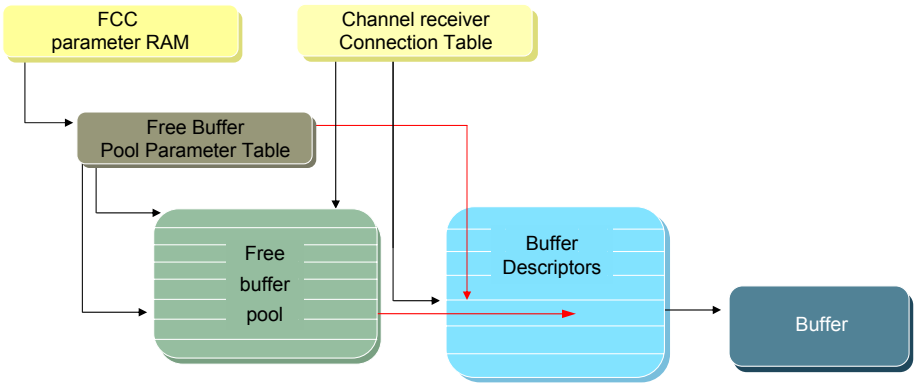
There can be two sets of receiver connection tables, one in dual port RAM, which is for channels zero through 255, and the rest in external memory for those channels above 255. However, there is a strange affect in how these are accessed, which must be accounted for when allocating memory. The offset to each connection table is the channel number multiplied by the connection table size, which is 32 bytes. The offset is applied to the base pointer which for the low channel numbers is the FCC parameter called int RCT base. This is also an offset from the top of dual port RAM.

The same calculation applies to the external table where the base pointer is another FCC parameter called ext RCT base. However, in this case, the first channel is 256 which means there must be a gap in the top of the external table from where the pointer relates to and the connection table entry for channel 256. Channel zero connection table entry is reserved and channel one entry is sometimes called the raw cell entry. This is related to the management function and will be described later.

An important consideration is that the channel relationship is only a mechanism to enable related cells to be handled appropriately. The user is free to utilize channels in the most convenient way to them. It makes sense to use the low order channels for fast, high priority connections because of the low latency for accessing internal RAM. If less than 255 connections are to be used, then low order channels can be used for all of them.

Global Buffer Allocation

- In the global allocation buffer mode, the CP obtains a receive buffer from a free buffer pool allocated by the user.
- It is useful for allocating memory among many ATM channels with variable data rates.



- When an ATM cell arrives, the CP opens the current BD in the ring, fetches a buffer pointer from the free BD pool associated with this channel, and writes the the pointer to the receive data buffer pointer field in the BD

One particular consideration is how much space must be allocated to receive buffers, and how efficiently it is used. If the application has well defined requirements for the quantity of data expected for all channels, then there is no problem defining the buffers for each channel. But what about the situation where it is not known exactly how the data is expected. The only safe way to allocate buffers might be to provide space for the maximum data flow that can be safely handled. But with many channels used this could require an enormous amount of memory and possibly much of it not actually used.

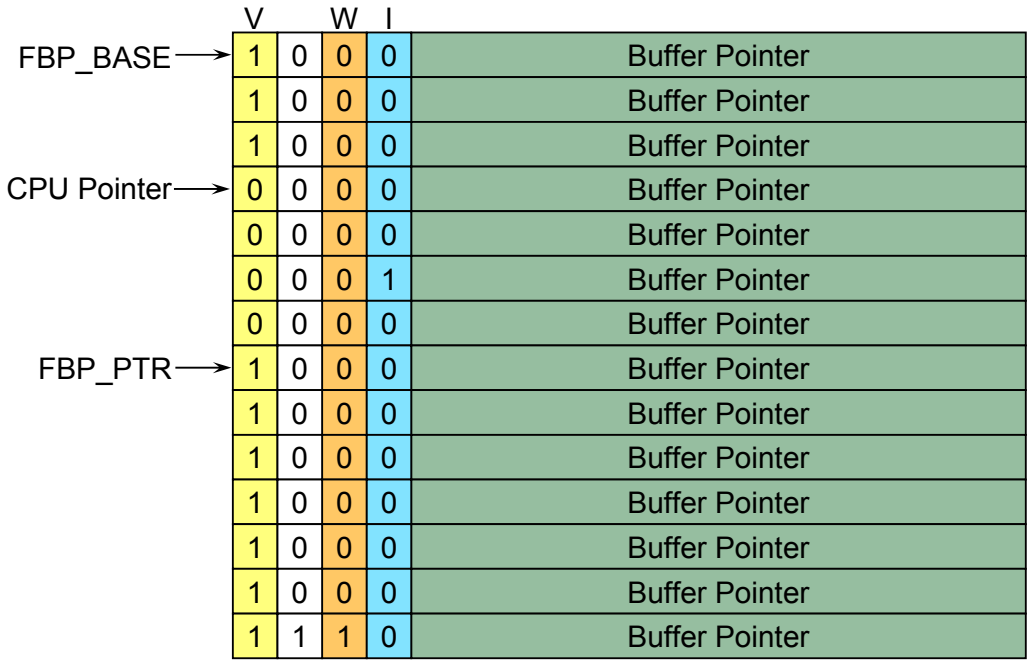
The alternative is to have what is known as global buffer allocation. This provides a dynamic allocation of buffers where a pool of buffers is allocated for general use, but not to specific channels. As a channel requires a new buffer, it is fetched from the free buffer pool and allocated to that channel. And as buffers become available they can be reallocated to the pool. This provides a much more efficient use of memory where buffers are only allocated as required.

To handle this mechanism there is an FCC parameter indicating a set of free buffer pool parameter table. It's possible to use up to four free buffer pools, and so there are a set of parameters for each. The parameters provide a pointer to the free buffer pool and a pointer to the current entry in the table. It also provides some control information, and the four most significant bits of the buffer pointer.

The free buffer pool is a table of pointers to the buffers and some control bits. Because the control bits require a proportion of the 32-bit entry those bits are provided by the parameter table. When a buffer is required, the channel receive connection table indicates if global allocation is used and, if so, which pool. The CP reads the FCC parameter to find the free buffer parameters, using the parameters indicated by the connection table, where it finds the buffer pointer. It then loads the buffer descriptor pointed to by the connection table with the buffer pointer from the free buffer pool concatenated with the four most significant bits from the parameter table.

Free Buffer Pool

- The free buffer pool provides the pointers to available buffers, but some information is also required for control. Indicators are required for **V**alid pointers, **W**here the end of the Table is, and when to generate an **I**nterrupt to service the table.


V
W
I
FBP_BASE
CPU Pointer
FBP_PTR
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
0 0 0 0 Buffer Pointer
0 0 0 0 Buffer Pointer
0 0 0 1 Buffer Pointer
0 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 0 0 0 Buffer Pointer
1 1 1 0 Buffer Pointer

Here is a representation of a free buffer pool. It basically consists of a table of buffer pointers and some control bits for handling it. The user must initialize the table with the pointers which are the base address of each buffer. The control necessary is an indication that the pointer is available, defined as the valid bit, an indicator for the end of the table defined as the wrap bit, and an interrupt bit to indicate where in the table an interrupt should be generated to service the table when necessary.

The table provides a rolling mechanism where the CPU must track buffer allocation and ensure that buffers are re-allocated as necessary. There is a pointer in the free buffer pool parameters to the first entry in the table, and also a pointer to the next available entry to be used. The user must provide a service pointer for the CPU to track buffer use. Initially, the free buffer pointer defined as FBP_PTR should be initialized to the same value as the base pointer, all the valid bits set to indicate that the buffer pointers are available, and the wrap bit set for the last entry in the table.

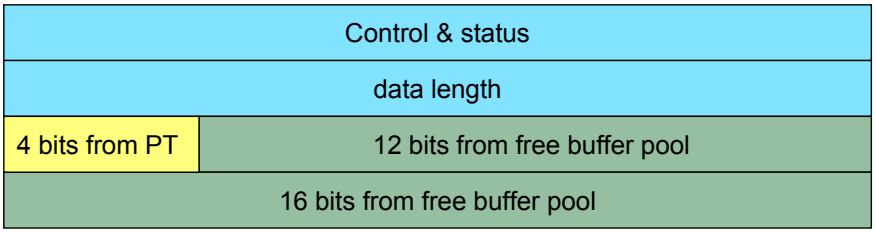
The interrupt bit should be set at a suitable location to indicate where the table needs servicing when the pointers are being used up. When a buffer is needed, the CP accesses the location indicated by the free buffer pointer and if the valid bit is set, then the buffer pointer is loaded into the buffer descriptor, concatenated with the 4 most significant bits from the parameter table. The valid bit is cleared indicating that that buffer is no longer available, and the free buffer pointer is incremented to the next table entry. This process will continue until it reaches an entry with the wrap bit set. That entry will be used, but then the processor will wrap back to the first entry by replacing the free buffer pointer with the base pointer.

Here is where we can see the requirement for the interrupt control function. The first function of the processor is to check the valid bit to ensure the buffer is available. In this case the bit would be clear because the first pass had used this buffer. Obviously, a service routine is required to handle this situation. That's where the interrupt bit comes in. When the CP encounters an entry with the interrupt bit set, a red line interrupt event is loaded into the interrupt queue to generate an interrupt to the CPU. The service routine must either set the valid bit for previously used buffers, if they are now free, or replace the buffer pointer with that of a free buffer and make it valid. The CPU service routine must provide it's own pointer to track this process and manipulate the interrupt bit as required. If the CP should encounter an entry with the valid bit clear, then there is a severe problem since data will be lost due to the lack of an available buffer. A bit will be set in the free buffer pool parameter table and a busy interrupt generated in the interrupt queue.

Buffer Pointer

- Because the free buffer pool is 32 bits wide, but requires three bits for control only 28 bits are provided from it for the buffer pointer, the remaining four bits are provided by the parameter table.

Buffer Descriptor

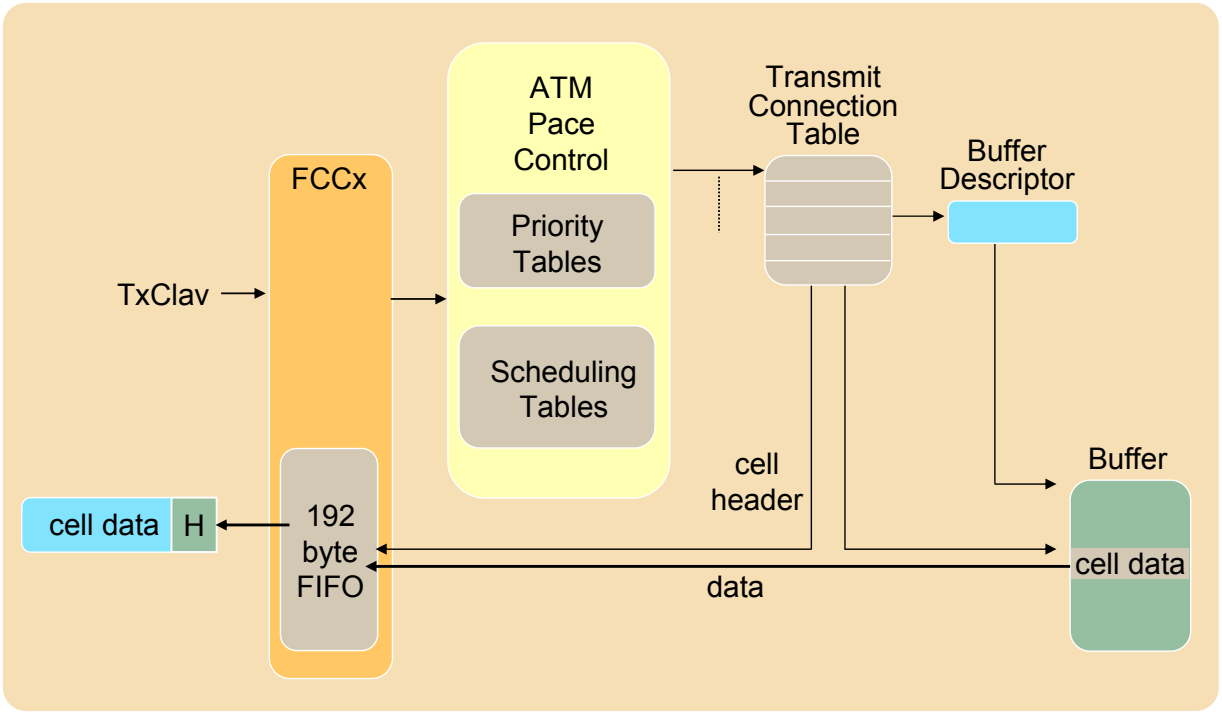


- It is possible to use up to four free buffer pools.
- The receiver connection table provides the selection of which pool to use.

The buffer descriptors for this function have the same format as for other protocols but because the free buffer pool table has only 28 bits available for the buffer pointer, the remaining most significant bits are supplied from the parameter table. The complete buffer pointer is used to deliver the cell data to the buffer.

ATM Pace Control

- Implements cell multiplexing by scheduling cells to be transmitted from various channels according to their frequency and priority.

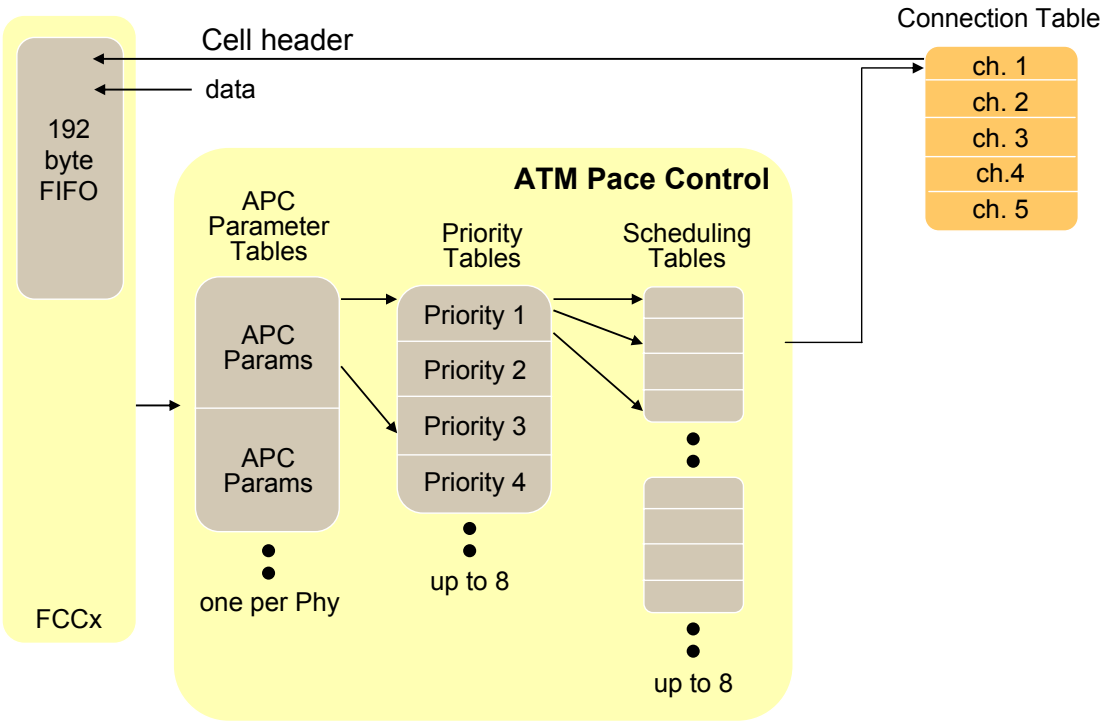


Having seen the basics of how the receiver handles incoming cells, now we'll examine the transmit function. Generally, for most protocols, this is the simpler function because the data to transmit is known and there is no concern for not having a buffer available. However, in ATM, there is the complication of which channels are scheduled for transmission, known as pace control.

The basic process is to determine which channel requires transmission next, find the appropriate cell data through the connection table and buffer descriptor, and load the cell into the FIFO, along with it's header. Because there is a relatively large FIFO of 192 bytes, then several cells are queued into it, to ensure a steady flow of cells to the PHY.

Once the next channel to be scheduled is determined, the connection table for that channel provides the buffer descriptor pointer and the offset to the appropriate buffer location for the next cell and the cell header, which is attached to the cell data, and is ready for transmission. The actual transmission is triggered by an input from the PHY called transmit cell available. As space becomes available in the FIFO when cells are transferred to the PHY, another cell is scheduled. Thus, the transmit cell available signal indirectly triggers the scheduling process.

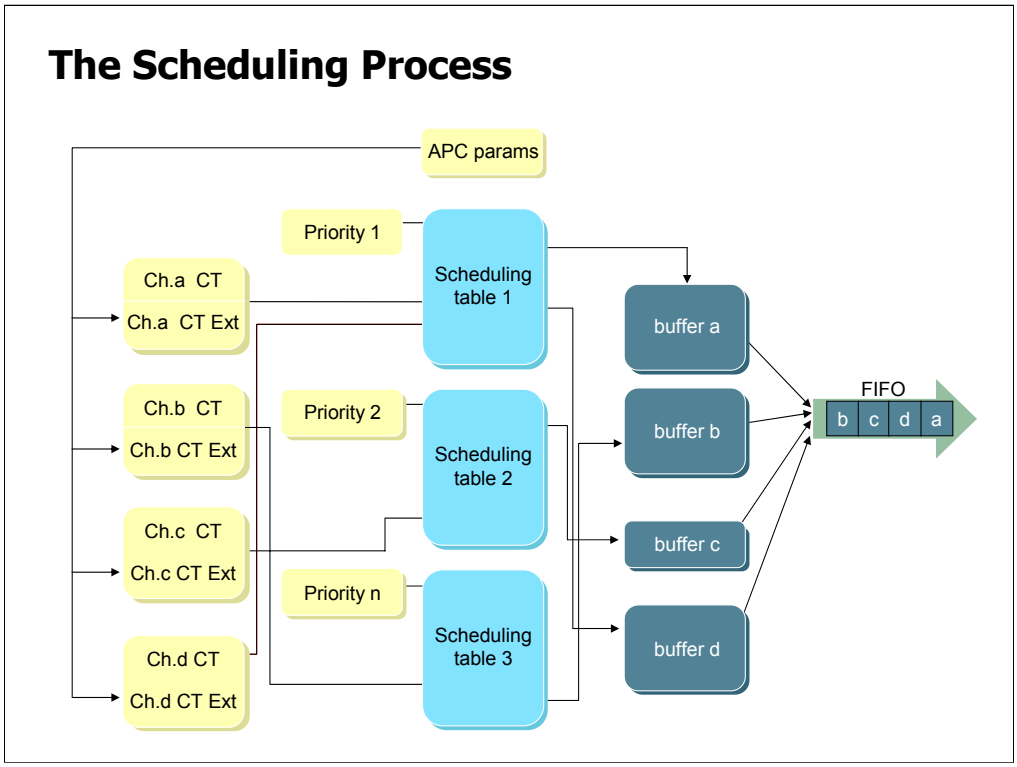
Basic Pace Control Structure



Pace control consists of pace control parameter tables, priority tables, and scheduling tables. For a multi PHY system, there is a pace control parameter table for each PHY. There can be up to eight priority tables and a scheduling table for each priority.

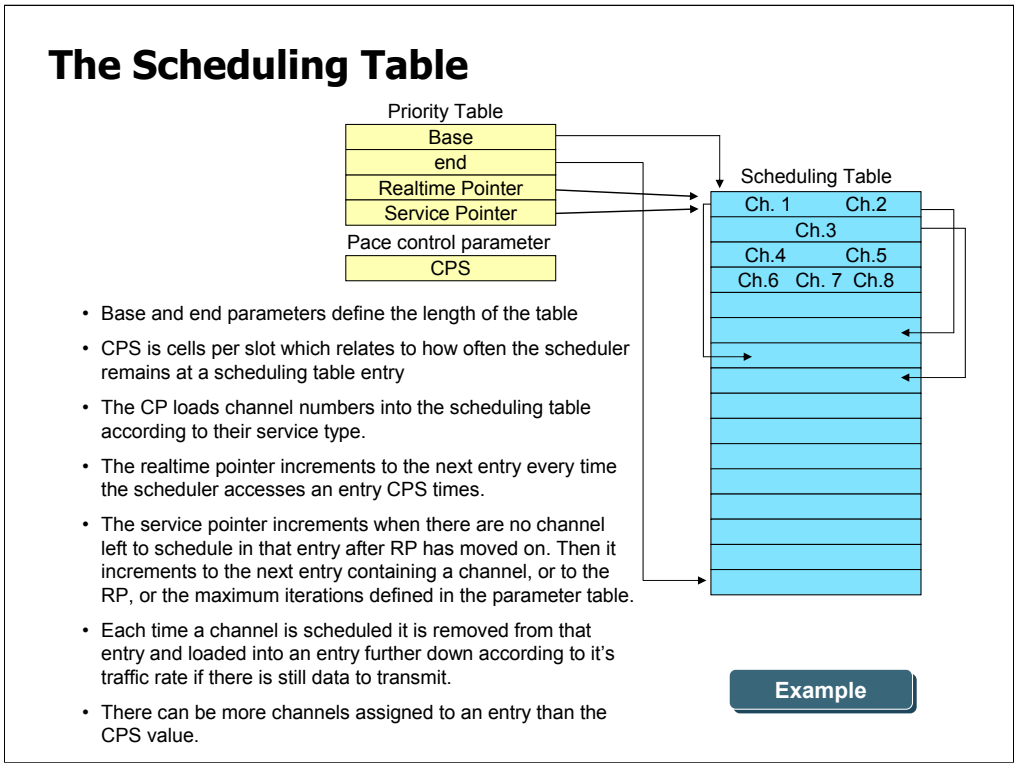
The parameter table contains pointers to the priority tables and information related to the type of service. The priority table contains the control pointers for the scheduling tables, and the scheduling tables define the channel to load into the PHY in order.

The user must set up the scheduling tables by defining the length of the table and scheduling rate, allocating the memory space, and initializing the control pointers. Once that is done, the tables are dynamically loaded by the CP from calculations performed using parameters defining the service requirements of the channels.



The scheduling table provides the organization of channels for transmission scheduling according to their transmission rate and priority, and for rescheduling subsequent transmissions. The priority table provides the means to identify where the user locates the table in memory and the length of the table.

The pace control parameters and extended connection tables provide the information for the communications processor to determine when and where to locate a channel number in the table. The priority table provides the pointers for the CP to access the table entries. As cells are required to be loaded into the FIFO, the scheduling tables determine the required channel to load and the connection tables provide the buffer location. The priority of a channel is defined by a command provided by the user in the Communications Processor Command Register known as the CPCR.


The Scheduling Table
Priority Table
Base
end
Realtime Pointer
Service Pointer
Pace control parameter
CPS
Scheduling Table
Ch. 1
Ch.2
Ch.3
Ch.4
Ch.5
Ch.6
Ch. 7
Ch.8
• Base and end parameters define the length of the table
• CPS is cells per slot which relates to how often the scheduler remains at a scheduling table entry
• The CP loads channel numbers into the scheduling table according to their service type.
• The realtime pointer increments to the next entry every time the scheduler accesses an entry CPS times.
• The service pointer increments when there are no channel left to schedule in that entry after RP has moved on. Then it increments to the next entry containing a channel, or to the RP, or the maximum iterations defined in the parameter table.
• Each time a channel is scheduled it is removed from that entry and loaded into an entry further down according to it's traffic rate if there is still data to transmit.
• There can be more channels assigned to an entry than the CPS value.
Example

One parameter called cells per slot controls how often the processor remains at each entry. To start, all pointers are initialized to the top of the table and as channels are enabled, the CP loads their numbers into the table. Each time there is room for a cell to be loaded into the FIFO, the next channel in the table is scheduled, then removed from that entry and reloaded into a subsequent entry according to it's traffic rate. The realtime pointer remains at the entry until the number of cell scheduling times reach the value of CPS and then it increments to the next entry. If there are no further channels to be scheduled in that entry, the service pointer also increments. But if more channels remain then the service pointer remains at that entry until all those channels are scheduled. Then the service pointer is incremented to the next entry with channels to schedule or to the realtime pointer, or by a value defined as the maximum iterations parameter.

This process works on the highest priority table first but if at any time there isn't a channel to send, then the process jumps to the next priority table. If at anytime no channels are loaded at any active entry, then an idle cell is sent. Rescheduling is performed as last in first out.

Cells Per Slot

The number of cells per slot is determined by the channel with the maximum bit rate which is achieved when the channel is rescheduled to the next table entry.

Maximum bit rate = line rate divided by cells per slot
so : Cells per slot = line rate divided by maximum bit rate

Example

A system has a PHY bit rate of 25.6 Mbps. with a maximum virtual channel bit rate of 2.56Mbps:

$$CPS = PHY / V_{MAX}$$
$$= 25.6 * 10^6 / 2.56 * 10^6$$
$$= 10$$

Now that we have an idea of how the process works, how do we get the values needed to set up the parameters for pace control? The cells per slot value to load into the pace control parameters is a function of line rate and the maximum channel bit rate. From the scheduling table, it's apparent that the fastest that a channel can be transmitted is when the channel is rescheduled to the next entry in the table. That way every time the scheduler moves to a new entry, it will be rescheduled. The parameters must therefore be initialized for the fastest channel.

As an example, consider a system where the line, or PHY, rate is 25.6 megabits per second and the fastest channel will operate at a maximum of 2.56 megabits per second. If the channel is rescheduled, every entry, 10 slots per second, will allow that rate.

Table Length

The number of entries per scheduling table is determined by the channel with the minimum bit rate and is achieved when the channel is rescheduled to the length of the table minus one .

Minimum bit rate = line rate / (cells per slot * (number of entries - 1))
so : Cells per slot * (entries - 1) = line rate / minimum bit rate

Example

Given the same system requirements for the previous example :
A system has a PHY bit rate of 25.6 Mbps
Maximum channel bit rate is 2.56Mbps
Minimum channel bit rate is 64Kbps

$CPS(M - 1) = PHY/V_{MIN}$ M = number of entries
$= 25.6 * 10^6 / 64 * 10^3$
$= 400$

so: $M = (400/CPS) + 1 = 41$

Having determined the cells per slot, the next item to define is how long is the scheduling table. The slowest a channel can be transmitted is when it is rescheduled the length of the table. In other words, if it is scheduled in the first table entry and then rescheduled to the last entry, then it will not be retransmitted until the scheduler has traversed the complete table length. In this case, the number of entries it is rescheduled is the table length minus one. This provides a minimum bit rate equal to the line rate divided by the combination of cells per slot multiplied by the table length minus one. From this the table length can be calculated.

Using the same example that was used to calculate the CPS value, and given that the slowest channel is to operate at 64Kbps; then the cells per slot times the table length minus one equals 400, which results in a table length of 41.

Service Categories

- The service categories relate traffic characteristics and QoS requirements to network behavior. Functions such as routing, Connection admission Control , and resource allocation are, in general, structured differently for each service category.
- Service categories are distinguished as being either real-time or non-real-time.

Service Type	Cell Rate Pacing	Priority
CBR	Peak cell rate	1
VBR-RT	Peak and sustain cell rate	2
VBR-NRT	Peak and sustain cell rate	3
ABR	Peak cell rate	4
UBR+	Peak and minimum cell rate	5
UBR	Peak cell rate	6

Acronyms
CBR: constant bit rate
VBR: variable bit rate
ABR: available bit rate
UBR: unspecified bit rate
RT: real-time
NRT: non-real-time

Additional parameters for VBR, ABR, and UBR

TCTE

To understand the remainder of the parameters to determine for setting up pace control, the service categories must be considered. The service categories relate traffic characteristics and quality of service requirements to network behavior. Routing, connection admission control, and resource allocation, are generally structured differently for each service category.

The service categories are as follows:

Constant bit rate which provides rigid bandwidth and low latency. This uses peak cell rate for pace control, and uses the highest priority.

Variable bit rate, which supports predictable data streams with defined average and peak traffic.

The difference between real time and non real time is that non real time is more tolerant of network delays. For this, peak and sustained cell rate is used for pace control using priority level 2 for real time and priority 3 for non real time.

Available bit rate supports burst traffic, using feedback mechanisms to manage the traffic. This uses peak cell rate with priority level 4.

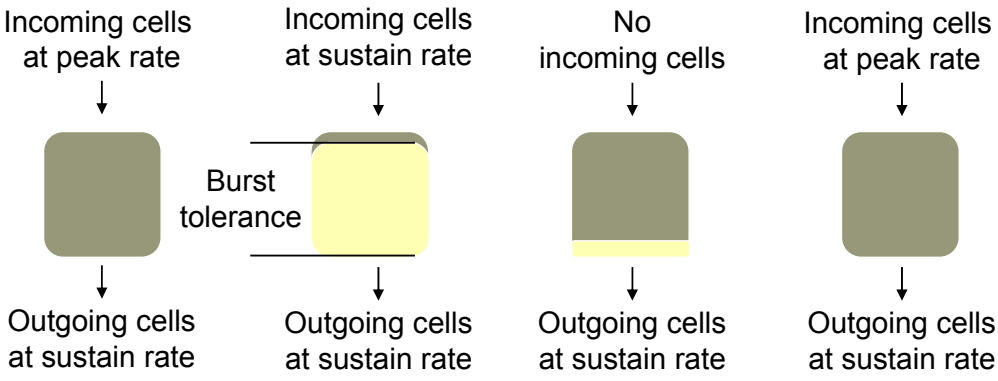
Unspecified bit rate has no traffic service guarantee. There are two options for this UBR plus which uses peak and minimum cell rate for pace control at priority level 5, while the lowest uses peak cell rate at priority level 6.

Service categories VBR; ABR; and UBR require additional connection table information, supplied in the Transmit Connection Table Extension.

Peak and Sustain Cell Rate

- Peak and sustain cell rate is bursting at peak rate for a burst tolerance limit followed by sustain rate after reaching the burst tolerance limit.
- This is the general cell rate algorithm (GCRA) often referred to as “leaky bucket algorithm.”

Example


Incoming cells at peak rate
Incoming cells at sustain rate
No incoming cells
Incoming cells at peak rate
Burst tolerance
Outgoing cells at sustain rate
Outgoing cells at sustain rate
Outgoing cells at sustain rate
Outgoing cells at sustain rate

Rescheduling

- When there is credit for burst transmission, the APC reschedules according to the peak rate; otherwise it reschedules according to the sustain rate.

If a channel is defined as peak cell rate traffic, then the pace controller schedules cells using parameters called PCR and PCR fraction from the transmit connection table. No other traffic parameters are used.

For variable bit rate, traffic can burst at the peak cell rate, but the average cannot exceed the sustained cell rate. The algorithm used is called the leaky bucket. To use the analogy, if you have an empty bucket with a small hole in the bottom and start filling it with water, as the water goes in it will also flow out through the hole. You can put the water in faster than it comes out, as long as the bucket is not full, but when it is full, you can only put water in at the rate the water flows out. If you slow down the input, then the bucket will start to empty, and so you can regulate the rate by inputting in bursts until it reaches a defined level, and then slow the rate down as needed.

In this case the water is cells to transmit, the flow rate out of the hole is defined as the Sustained Cell Rate parameter, SCR, the burst fill is the Peak Cell Rate parameter, PCR, and the defined level to stop bursting is the Burst Tolerance parameter, BT. The concept is that cells can be scheduled at the peak cell rate until the burst tolerance is reached, at which time they must only be scheduled at the sustained rate. So, when the cells are transmitted at less than the burst tolerance, they will be rescheduled in the table at the peak rate defined in the connection table. When they reach the burst tolerance, they will be rescheduled in the table at the sustained cell rate.

The connection table defines the traffic type in a parameter called ATT. If this is defined as peak cell rate, no extended connection table pace control parameters are used. If defined as peak and sustained cell rate, the extended connection table parameters for SCR and SCR fraction and BT are used.

Peak Cell Rate - PCR

- PCR is the reciprocal of the minimum spacing of cells of an ATM connection on a transmission link.
- Peak cell rate = line rate divided by (channel rate multiplied by the number of cells per slot)

$$PCR = Line\ Rate\ /(VC\ rate\ * CPS)$$

Example

Given a virtual channel uses 6.00 Mbps of a 155.52 Mbps line rate and the CPS = 8.

$$PCR = 155.52\ * 10^6\ /(6\ * 10^6\ * 8) = 3.24$$

TCT Values

There are two values to be initialized for peak cell rate:
TCT[PCR] and TCT[PCR_FRACTION].

Therefore: TCT[PCR] = 3
and TCT[PCR_FRACTION] = 0.24*256/256
= 61.44/256 ~ 62/256
= 62

The peak cell rate is the reciprocal of the minimum spacing of cells in a connection. In other words, the higher the rate the closer they must be scheduled. To calculate the appropriate value for peak cell rate, the channel rate required multiplied by the cells per slot is divided into the line rate. The rates are in bits per second and VC stands for virtual channel. The reason that the cells per slot are used in the calculation is that a channel can only be scheduled once per table entry and if there is more than one cell per slot, then that will slow down the rescheduling of that channel.

For example, if the channel is rescheduled every entry but there are two cell per slot, then the channel can only be transmitted every other transmission, and will have a rate of half the line rate. To distinguish between slots and entries, the entries refer to the pace control table entry where the channel numbers are loaded by the scheduler. The slot refers to the number of cells placed on the line before the controller advances to the next entry in the table. There can be more channels in an entry than cells per slot.

Now, given this formula, the PCR value for the controller can be calculated. To demonstrate it lets look at an example. Given a virtual channel rate of 6 megabits per second, a line rate of 155.52 megabits per second, and 8 cells per slot, the resulting PCR is 3.24. This must be loaded into the transmit connection table, but obviously the fractional part requires special treatment. That is handled by providing an integer as a proportion of 256. This results in the value 61.44, which must be rounded up to 62. Thus the values programmed are PCR is three and PCR fraction is 62.

Calculation for Sustain Cell Rate

Example

Given a line rate of 155.52 Mbps a virtual channel uses a peak cell rate of 6 Mbps, sustained cell rate of 2 Mbps, a maximum burst size of 1000 cells, and the CPS = 8.

Sustained Cell Rate

$$SCR = \text{Line Rate} / (\text{VC rate} * \text{cells\_per\_slot})$$
$$= (155.52 * 10^6) / (2 * 10^6 * 8)$$
$$= 9.72$$

$$SCRF = 0.72*256/256$$
$$= 184.32/256 \sim 185/256$$
$$= 185$$
$$=$$

Burst Tolerance

$$BT = [(MBS - 2)*(SCR - PCR)] + SCR$$
$$= [(1000 - 2)*(9.72 - 3.24)] + 9.72$$
$$= 6476.76$$

TCTE values

SCR = 9
SCRF = 185
BT = 6477

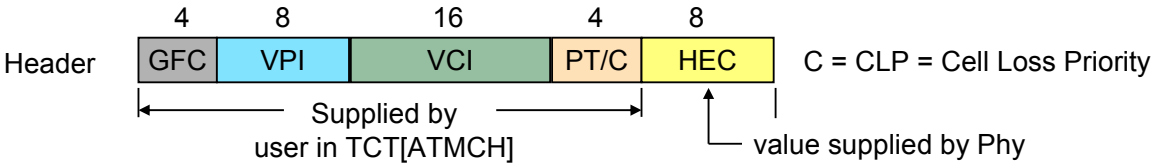
For peak and sustained cell rate traffic the PCR is calculated as shown before. The sustained rate value is calculated in the same way, but uses the channel rate defined as the sustained cell rate. It also requires a burst transfer rate.

For an example, consider the same situation as the previous example but this time the sustained rate is defined as two megabits per second with a maximum burst size of one thousand cells. Using the same formula to calculate SCR as for PCR results in a value of 9.72. The fraction is also calculated in the same way, resulting in a rounded up value of 185. These values are programmed into the transmit connection table extension parameters, resulting in SCR being 9 and SCR fraction being 185.

The burst tolerance formula is given as maximum burst size minus two, multiplied by the difference between peak and sustained rate, and adding the sustained cell rate to the result. With the given maximum burst size of 1,000 this results in a value rounded up to 6,412.

Cell Loss Priority

- Cell Loss Priority control: For some service categories the end system may generate traffic flows of cells with Cell Loss Priority (CLP) marking. The network may follow models which treat this marking as transparent or as significant. If treated as significant, the network may selectively discard cells marked with a low priority to protect, as far as possible, the QoS objectives of cells with high priority.



Rescheduling and CLP

Using VBR, scheduling is affected by CLP in one of two ways:

1. Regular: CLP = 0+1 cells are rescheduled by PCR or SCR according to the GCRA state (“leaky bucket”) .
2. VBR Type 2: CLP = 0 cells are rescheduled as above. CLP = 1 cells are rescheduled by PCR.

There is another feature that affects transmission rate for VBR traffic. Within the cell header is a four bit parameter designated as PT/C. The C portion stands for cell loss priority. This is a single bit where the user can indicate the importance of the cell. If the cell loss priority feature is turned on, then when the system gets overloaded and there is the potential that cells will be lost, then those with this bit clear will be discarded first.

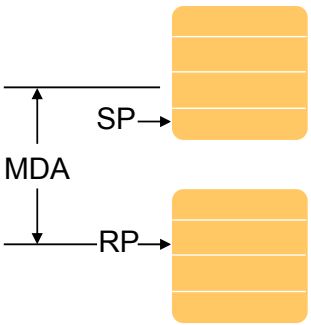
Peak and Minimum Cell Rate

- If the delay in service of a priority level is greater than Maximum Delay Allowed the controller reschedule channels in this priority level according to the Minimum Cell Rate parameter. If the delay is less than MDA then it will re-schedule channels in this priority level according to PCR.
- This reduces the rate for the UBR service to improve the higher priority channels.

Examples

SP→ RP < MDA

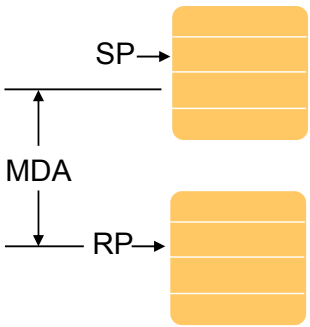
Scheduling Table



Pointers are within maximum delay allowed then schedule at peak cell rate

SP→ RP > MDA

Scheduling Table



Pointers are outside maximum delay allowed then schedule at minimum cell rate

Unspecified Bit Rate, UBR, uses peak and minimum cell rate for rescheduling cells for transmission. In this case, a maximum delay allowed is specified to track the service and real pointer to the scheduling table. The scheduling rate is modified depending on whether the distance between the pointers is less than or greater than the maximum delay allowed MDA.

As seen earlier, depending on how many channels are in each entry of the table, the real pointer can move ahead of the service pointer. The more channels in an entry, the further the real pointer can get ahead. This could result in problems of some channels not getting scheduled often enough. The solution is to check the distance between the pointer and if the distance becomes too great, then reduce the rescheduling rate for the UBR channel, which has a lower priority, resulting in better performance for the other channels. When the problem is resolved so that the distance between pointers is acceptable, then the rescheduling rate for the UBR channel is increased.

Minimum Cell Rate

The method of calculating the minimum cell rate value is the same as peak cell rate

MCR = line rate divided by (channel minimum rate multiplied by the number of cells per slot)

MCR = Line Rate /(VC rate * CPS)

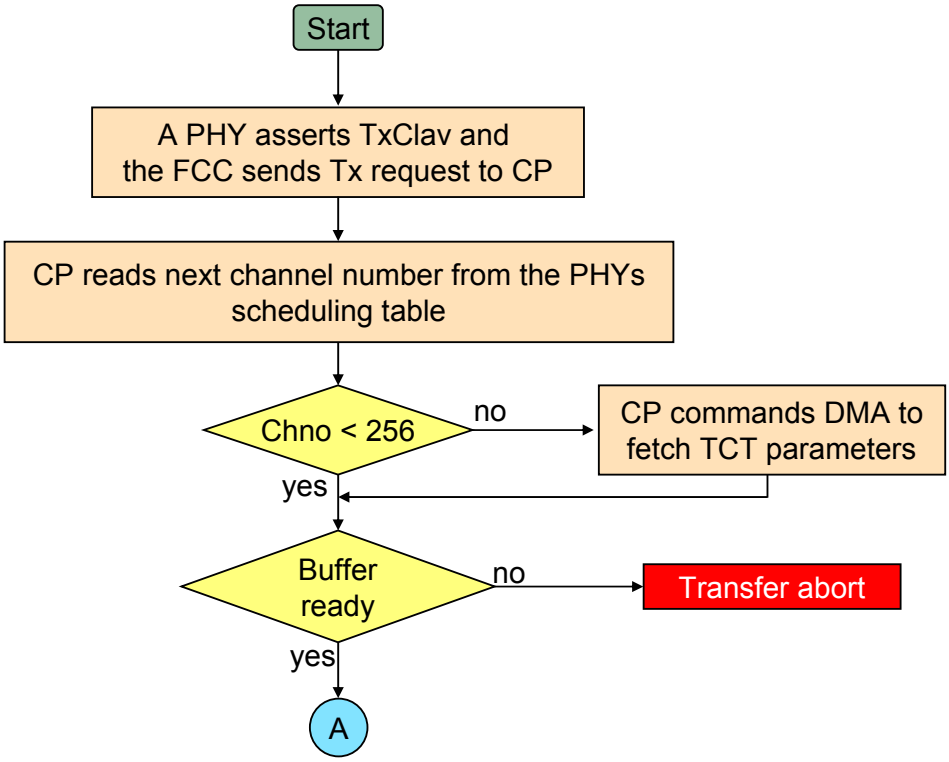
Example

Given a line rate of 155.52 Mbps a virtual channel uses a minimum cell rate of 1 Mbps, and the CPS = 8.

MCR = Line Rate /VC rate * cells_per_slot
= (155.52 Mbps/(1 Mbps * 8)
= 19.44
MCR = 19
MCRF = 0.44*256/256
= 112.64/256 ~ 113/256
= 113

The minimum cell rate parameter value is calculated in the same way as peak and sustained rates, the difference being the rate defined for the channels minimum rate. Using a similar values as for the previous examples, with a line rate of 155.52Mbps per second, a minimum cell rate of 1Mbps, and 8 cells per slot, the MCR value is 19 with a fraction of 113.

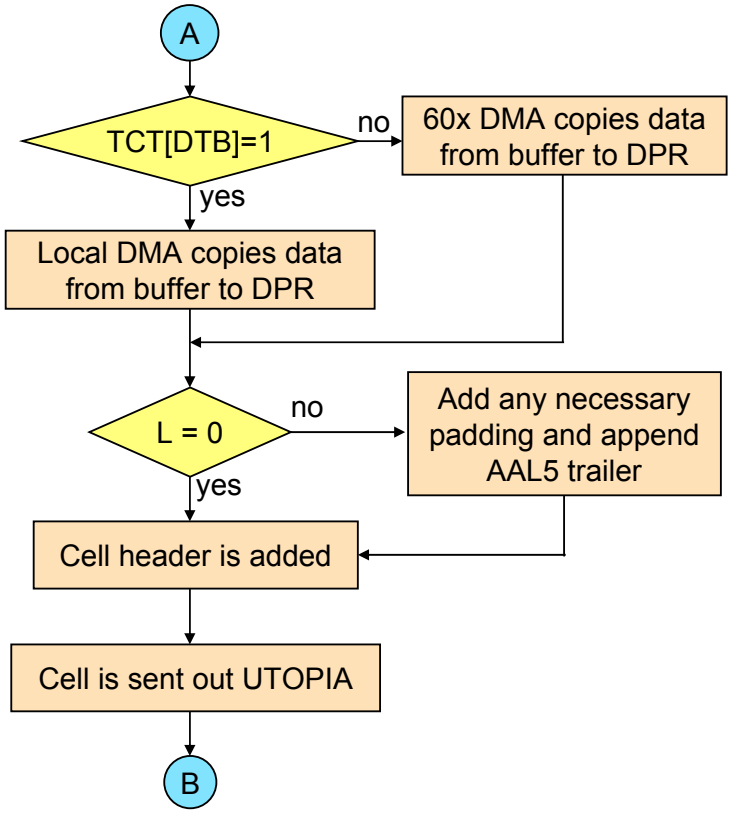
How the FCC Transmits a Cell (1 of 3)



The previous discussion indicates the basics of how ATM cells are transmitted and how scheduling is handled. Here is a basic flow chart of how the transfer is handled. Initially, a transfer is indirectly instigated by the PHY, indicating that it has room for a cell. In fact, the cell to be transferred next is already in the FIFO, but this activity will result in a cell being fetched to fit the next available space in the FIFO.

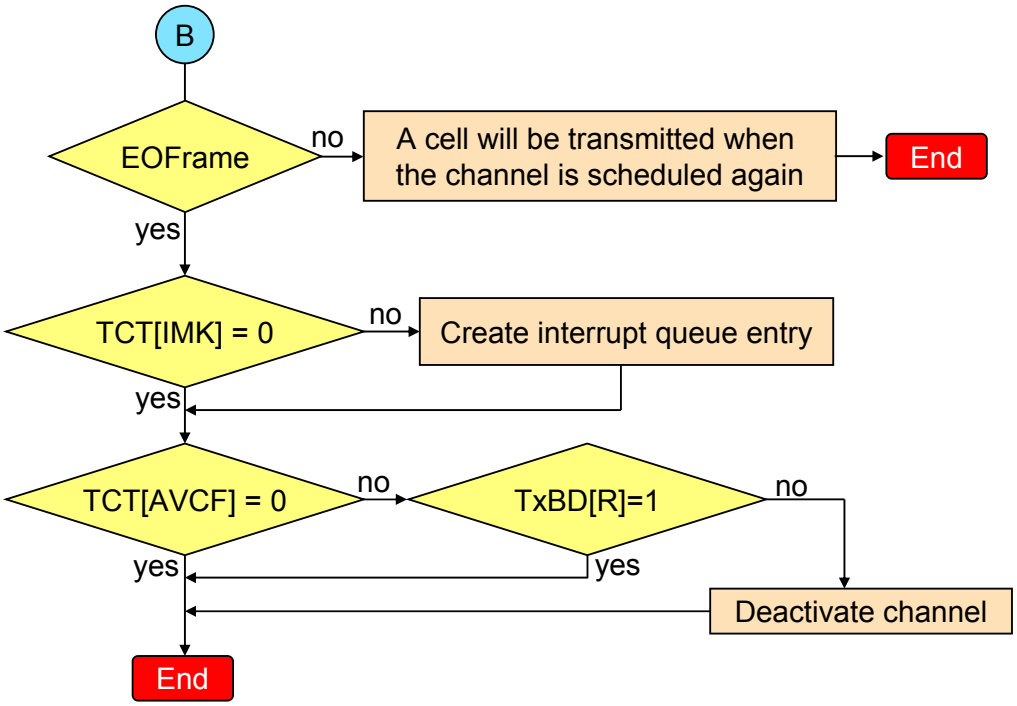
The comms processor reads the next channel number from the scheduling table, and must now read the transmit connection table, but where is it ? Well, if the channel number is less than 256, then the TCT is in internal memory and no external access is required. But if it's greater than 255, then it's in external memory and a DMA transfer is necessary to access the connection table. Once the connection table is accessed, then the buffer descriptor location is provided where there is an indication, if a buffer is ready. If there is no buffer ready to transmit, then the transfer is aborted and an idle cell is indicated for that slot. The flow chart continues on the next page,

How the FCC Transmits a Cell (2 of 3)



Assuming that a buffer is ready, the buffer descriptor provides the buffer pointer. The connection table parameter, DTB, indicates whether the data buffer is on the 60x bus (system bus) or the local bus, and the cell data is transferred into dual port RAM, used as an intermediate buffer before the FIFO. If this is an AAL5 cell, then a check is made for the end of frame and if it's not, then the cell header is added to the cell data in the dual port RAM area from the connection table. If, however, this is an AAL5 cell and is the last in the frame, then the AAL5 trailer is appended and any necessary padding is added if the data is not 48 bytes long. Then the cell is forwarded to the FIFO, and eventually it will be transferred on the UTOPIA to the PHY on it's turn. The flow chart continues on the next page.

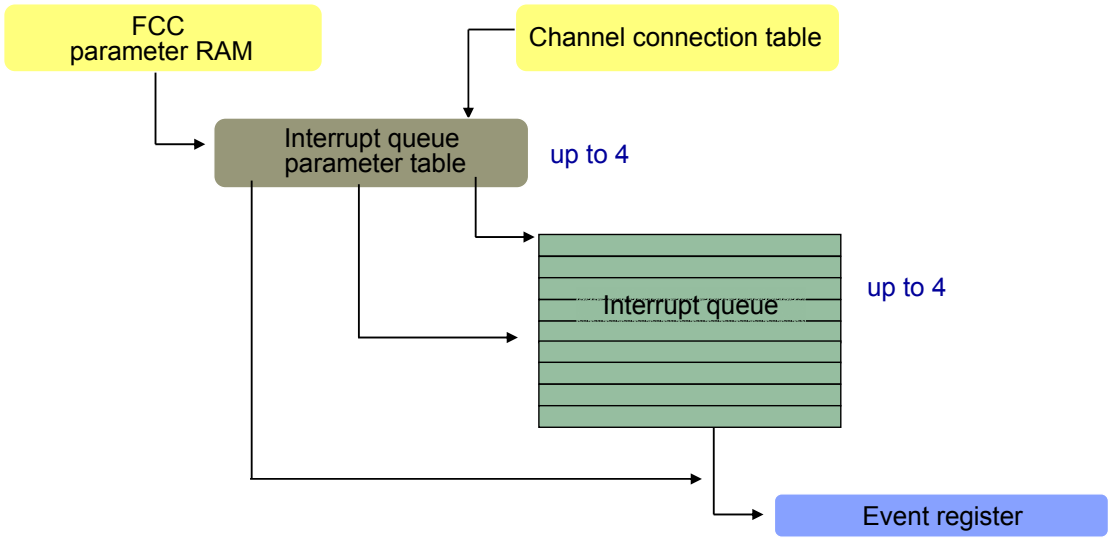
How the FCC Transmits a Cell (3 of 3)



If this cell is not the last cell in the frame, then this process is complete and will be repeated on the next occasion that this channel is encountered in the scheduling table. However, if this is the last cell of the frame, then the IMQ parameter, which is the interrupt mask bit, determines if an event is entered into the interrupt queue for the end of frame. If the auto VC off is selected and there are no buffer descriptors ready for this channel, then the channel will be deactivated and will no longer be rescheduled.

Interrupt Queue Organization

- With many channels but only one event register the Interrupt queue is used to track events for each channel.
- The connection table selects which of 4 queues is used, and parameters define where they are.



- An entry loaded into the queue triggers a global event in the Event register, controlled by a count parameter. The service routine examines the queue for event information.

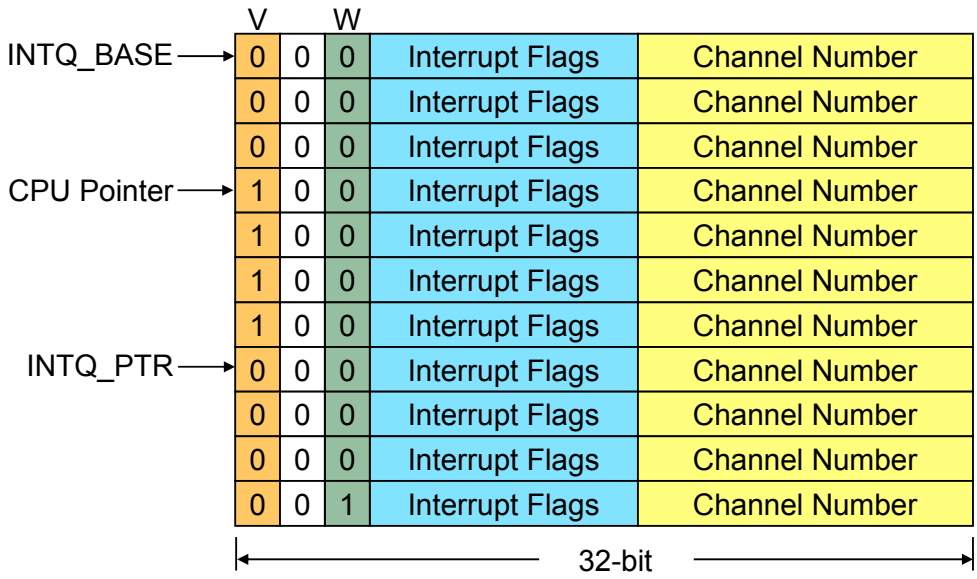
As with all of the controllers, the FCC generates events to indicate functions that might require servicing. What happens in most cases is that an event bit is set in the event register, and if unmasked, it can result in an interrupt to the core. However, in this case the events relate to channels and since there are many channels possible and only one event register the process is dealt with differently. There is still an event register that is used to signal the Interrupt request, but to handle each channel, the specific event is written to an entry in the interrupt queue, which must be set up by the user in memory. The interrupt queue entry effectively replicates the event register of the simpler controllers, but also contains some control bits.

The user must setup the queue in memory by allowing the space, defining the base address of the queue in the Interrupt parameter table, which must also have a pointer to it's base in FCC parameters. There must also be a current entry pointer which the user initializes to the top of the queue, after which time the controller handles it automatically.

Because there could be very many events which could result in numerous interrupts, there is also a counter in the queue parameter table that allows the user to delay setting the global interrupt bit in the event register until a pre-determined number of events have occurred. That reduces the number of interrupts generated, and several events are serviced for each interrupt. Up to four different interrupt queues can be used, providing the user with a variety of options on how the interrupts can be handled for different channels. The choice is made in the connection table.

Interrupt Queue

- The interrupt queue is a structure in memory in which event information is stored when a channel interrupt occurs.


V
W
INTQ_BASE
CPU Pointer
INTQ_PTR
0 0 0 Interrupt Flags Channel Number
0 0 0 Interrupt Flags Channel Number
0 0 0 Interrupt Flags Channel Number
1 0 0 Interrupt Flags Channel Number
1 0 0 Interrupt Flags Channel Number
1 0 0 Interrupt Flags Channel Number
1 0 0 Interrupt Flags Channel Number
0 0 0 Interrupt Flags Channel Number
0 0 0 Interrupt Flags Channel Number
0 0 0 Interrupt Flags Channel Number
0 0 1 Interrupt Flags Channel Number
32-bit

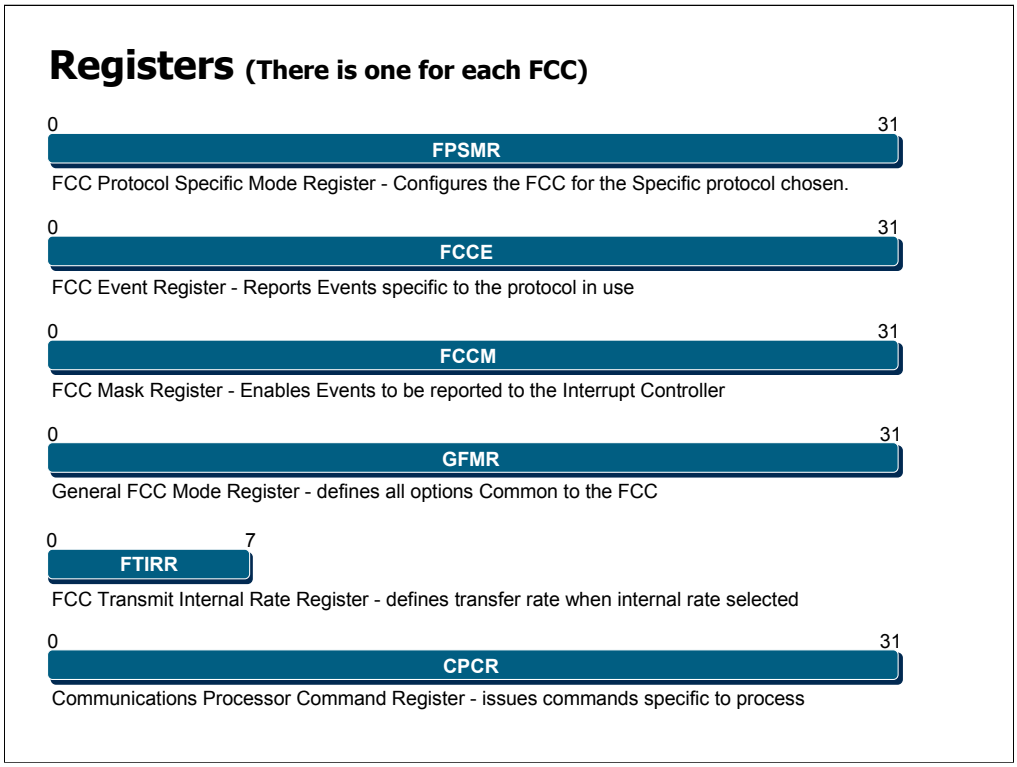
- When a channel interrupt occurs, CP writes the interrupt information to the location pointed to by INTQ_PTR.
- Entries with the valid bit set need to be processed by the CPU.
- The end of the queue is marked with the W = 1.

When the event is written into the entry, the channel number that it relates to is written with it. A valid bit is required for the controller to recognize that an entry is available to be used, and a wrap bit to indicate where the end of the queue is. Initially, all the queue entries should be cleared, except for the wrap bit in the last entry of the queue, which indicates the last entry where the controller will wrap back to the first entry after using it, thus forming a circular queue.

When the first event occurs, the controller accesses the entry pointed to by the interrupt queue pointer, which at that time should be at the top of the queue. It examines the V bit and if it is clear, then the entry can be used. The specific event bit is set in the interrupt flags region, the channel number associated with the event is loaded into the lower 16 bits of the entry. The V bit is set to indicate that this entry now contains valid information for the interrupt service routine. The interrupt queue pointer is incremented to the next entry and the associated global interrupt bit is set in the event register. If the associated bit is set in the mask register, the global interrupt bit set will signal an interrupt request to the interrupt controller.

As events are generated, the controller works down the queue, filling entries, until it reaches the entry with the wrap bit set. After using that entry, the controller wraps back to the top of the queue. Obviously, the events entered into the queue must be serviced. The global interrupt in the event register should result in a CPU interrupt, which will cause a service routine to examine the available information to deal with it. The FCC service routine will use the entries in the queue to service the events, and for all events serviced, must clear the V bit to enable those entries to be reused for subsequent events. For this purpose, the service routine must maintain a CPU pointer to enable it to track where to service next.

If at any time an event occurs and the queue entry pointed to by the interrupt pointer is not available, that is the valid bit is set, then an interrupt queue overflow event is set in the event register. This is, of course, a very serious problem because now information has been lost, so this must be guarded against by ensuring that the queue is serviced appropriately.

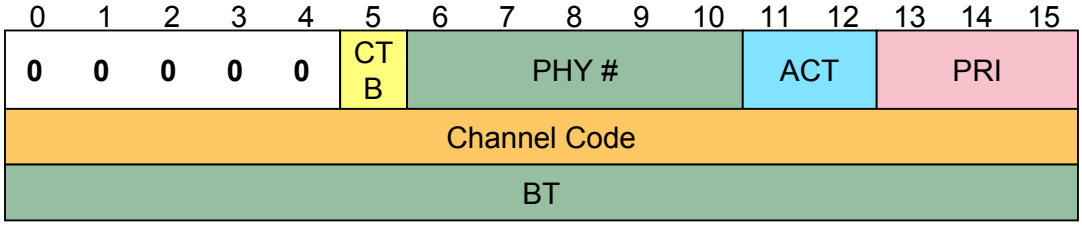

Registers (There is one for each FCC)
0
31
FPSMR
FCC Protocol Specific Mode Register - Configures the FCC for the Specific protocol chosen.
0
31
FCCE
FCC Event Register - Reports Events specific to the protocol in use
0
31
FCCM
FCC Mask Register - Enables Events to be reported to the Interrupt Controller
0
31
GFMR
General FCC Mode Register - defines all options Common to the FCC
0
7
FTIRR
FCC Transmit Internal Rate Register - defines transfer rate when internal rate selected
0
31
CPCR
Communications Processor Command Register - issues commands specific to process

The registers are common to the FCC but FPSMR, FCCE and FCCM are protocol specific and so have to be treated differently for the protocol in use. GFMR is only required in ATM mode for selecting the protocol and enabling receive and transmit. FTIRR is a register only used in ATM mode, and is only available on FCC1 and FCC2. It is used to select internal rate transmission, which schedules according to a timer and provides the timer initial value. Each FCC has four of these registers, one for each of the first four PHYs. When in slave mode, only FTIRR0 is used.

ATM transmit commands

- Before issuing a transmit command for an ATM channel all initialization must be complete.
- For the command the following information must be in the FCC parameters.

COMM_INFO - in Parameter RAM at offset 0x86



CT	connection table bus
PHY	PHY number
ACT	select for VBR channel
PRI	APC priority
Channel code	channel number for this command
BT	Burst tolerance

Reference Manual section 29.14

Once a channel has been initialized and is ready to transmit, the transmit command must be sent to the communications processor. For this the parameters shown here must be in the COMM_INFO of the FCC parameters.

CT defines which bus must be accessed for the external connection table.

PHY indicates which PHY the channel is to be transferred to in multi PHY mode.

ACT defines if this is a VBR channel or some other type.

PRI defines the pace control priority.

BT defines the burst tolerance for a VBR channel.

Sending the ATM transmit command in the command register results in this channel being loaded into the pace control scheduling table defined by the priority field of these parameters.